B1

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 1/00	A1	(11) International Publication Number: WO 95/35533 (43) International Publication Date: 28 December 1995 (28.12.95)

(21) International Application Number: PCT/CA95/00354

(22) International Filing Date: 16 June 1995 (16.06.95)

(30) Priority Data:
08/261,496 17 June 1994 (17.06.94) US

(71) Applicant: MEGALODE CORPORATION [CA/CA]; 935 Sheldon Court, Burlington, Ontario L7L 5K6 (CA).

(72) Inventors: PENKAVA, Josef; 1005 Upper Gage Avenue, Hamilton, Ontario L8V 4L2 (CA). CLARK, Robert, C.; 185 Seaton Street, Toronto, Ontario M5A 2T5 (CA). SIRBU, Victor; Apartment 7, 358 Clendenan Avenue, Toronto, Ontario M6P 2X4 (CA). LUNDY, Douglas, H.; 2182 Maplewood Drive, Burlington, Ontario L7R 2C5 (CA). CONFORZI, David, J.; 720 Courtland Place, Burlington, Ontario L7R 2M6 (CA). MAXWELL, W., Norman; R.R.#3, Puslinch, West Flamborough, Ontario N0B 2J0 (CA).

(74) Agent: MACRAE & CO.; Station B, P.O. Box 806, Ottawa, Ontario K1P 5T4 (CA).

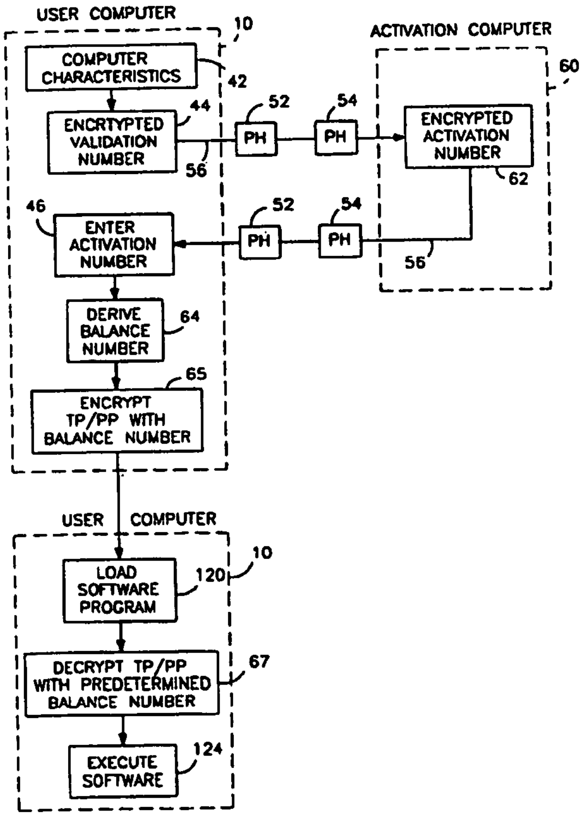
(81) Designated States: AM, AT, AU, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LT, LU, LV, MD, MG, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TT, UA, UG, UZ, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG), ARIPO patent (KE, MW, SD, SZ, UG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: METHOD FOR PREVENTING USE OF SOFTWARE ON AN UNAUTHORIZED COMPUTER

(57) Abstract

A method for preventing use of software on an unauthorized computer. The software is programmed to encrypt and output to the user a validation number derived from information received by the software from the computer of one or more computer characteristics providing an unchangeable and unique computer identification. A second computer is operated for the software vendor to encrypt an activation number derived from the validation number and supplied to the user for input to the user's computer. The activation number includes one or more randomly-generated digits which, when a predetermined mathematical operation is performed thereon and on at least one of the digits of the validation number, yields a derived balance number. A preselected signature and other information is randomly scattered among randomly generated bytes along with a product identification number as a thumbprint/productprint which is encrypted by the balance number derived by the user's computer from the validation and activation numbers and which is on the hard disk drive of the user's computer. The software is authorized for use in the user's computer if the preselected signature is retrieved after the predetermined balance number is applied to decrypt the information including the preselected signature.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria
AU	Australia
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CS	Czechoslovakia
CZ	Czech Republic
DE	Germany
DK	Denmark
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgystan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakhstan
LI	Liechtenstein
LK	Sri Lanka
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SI	Slovenia
SK	Slovakia
SN	Senegal
TD	Chad
TG	Togo
TJ	Tajikistan
TT	Trinidad and Tobago
UA	Ukraine
US	United States of America
UZ	Uzbekistan
VN	Viet Nam

METHOD FOR PREVENTING USE OF SOFTWARE ON AN UNAUTHORIZED COMPUTER

The present invention relates generally to the prevention of unauthorized use of software. More particularly, the present invention relates to the preventing of a computer program from being executed on a computer system or computer network, other than one which has been previously authorized.

As computer systems and software have proliferated, the problem of software piracy has also increased. A computer program is typically installed in a computer with a fixed disk or hard drive by transferring the program from a floppy disk or CD-ROM (purchased from a software publisher) to the fixed disk for subsequent use by the computer system. While the program may have originally been legitimately purchased, the purchaser may thereafter make copies for use by the purchaser or others in other computer systems or may simply use the floppy disk to install the software in other computer systems, without permission of the software publisher, thereby depriving the software publisher of the additional revenues of sale of additional software packages to which the publisher is entitled. Although back-up copies of software are normally considered desirable, it is also desirable for the financial health of the software industry that such "piracy" be stopped.

Various techniques have been proposed for elimination or reduction of software piracy. Many of these techniques are described in U.S. patent 5,113,518 to Durst, Jr. et al, which is incorporated herein by reference. Unfortunately, these techniques have met with only limited success as procedures have been found by the "pirates" or "hackers" for circumventing these techniques. Some of these techniques may also be so inconvenient as to deter the customer from purchasing the software. For example, one such technique utilizes hardware in the form of a device called a "dongle" which is connected to the computer, and the software must confirm the presence of this device by means of a coded response before it can be activated. Such a device is described in U.S. patents 4,446,519; 4,562,306; 4,685,055; and 5,182,770. However, this undesirably requires the purchase by the customer of such a device for each new software package which is purchased. Further, this technique can be defeated by discovery the correct coded response and providing it through a modification of the program.

As pointed out in Durst, Jr. et al, the problem is not so much in the act of copying the software package as it is in the use of copies of the software on various computer systems without compensating the publisher for the right to use those copies. It is therefore a primary object of the present invention to prevent a computer program from being executed on an unauthorized computer system.

Durst, Jr. et al discloses a technique for preventing a computer program from being used by a computer system other than a designated system. The values of certain characteristics exhibited by the designated computer system first are stored, and then the values of those same characteristics exhibited by the computer system which is intended to use the computer program are measured and compared to the stored values. If the compared values are substantially the same, the computer program may be executed. However, if they are different, the computer system which was intended to use the program is inhibited from executing that program. These characteristics are disclosed to be one or more, and preferably at least two, of the following: an identification of the processor included in the computer system, the clock speed of the clock generator included in that system, an identification of the ROM normally provided with the processor, the wait time assigned to the processor for accessing a RAM, the actual rotary speed of a disk drive normally provided with the computer system, the access speed of that disk drive, and the sector interleave value of that disk drive.

Inherent characteristics such as proposed in Durst, Jr. et al have a tendency to have no current uniqueness (although perhaps unique at one time, standardization may have resulted in non-uniqueness, for example, there is now a standard disk drive speed), or the characteristics may change over time undesirably making the software unavailable on the computer on which it is originally installed. Furthermore, the values of the characteristics are stored in the software to be compared with the values of characteristics of a computer on which the software is to be used. This has the disadvantage of being easy to circumvent since the values are stored in a known location in all programs, thus being accessible to every level of programmer.

U.S. patent 4,740,890 to William discloses the use of a remote computer to provide unlocking codes derived from master lists or algorithms. In the field of radio-frequency transmission of data, data security has been maintained by the use of coded transmission utilizing a pair of numbers wherein a plurality of randomly-generated digits in one number has a mathematical relationship to the other number so as to yield a prime number for coding the transmission, and the same prime number is used for decoding the transmission.

As discussed in U.S. patent 4,319,079 to Best, various encryption systems have been developed to provide data security within data processing systems. However, computer-aided techniques for breaking codes are becoming more sophisticated.

Techniques have also been proposed for activating software remotely. For example, U.S. patent 5,222,134 to Waite et al discloses such a technique wherein a computer is provided with a registration shell, and a data link is established between the computer and a registration computer. By providing the registration computer with various information, a potential licensee can register to utilize the program. Once the registration process is complete, a tamper-proof overlay program is constructed at the registration computer and transferred to the user's computer. The overlay includes critical portions of the main program, without which the main program would not operate. This process undesirably requires a modem on the user's computer.

U.S. patent 5,199,066 to Logan, which is incorporated herein by reference, discloses a method and system for protecting a software program recorded within a storage medium for use with or transmission to computer or processor based hardware. A hardware code uniquely associated with the particular hardware and a first software code uniquely associated with the particular embodiment of the software are inputted. The hardware code is stated to be the numeric serial number of the hardware upon which the program is to operate. It is further stated that, in the case of some computers and some storage media, the program may have the ability to recall or otherwise obtain and input the software serial number and possibly the hardware serial number without any specific action by the user. A first predetermined operation is performed upon the hardware code and the first software code to produce an intermediate code. A unique activation code obtained from the software supplier is inputted and a second predetermined operation is performed upon the intermediate code and the activation code to produce a second intermediate code. The second intermediate code is compared to a second software code uniquely associated with the particular embodiment of the software and stored in a hidden location within the software. The use of the software is enabled only if the second intermediate code and the second software code are identical. By a formula, the hidden software code changes each time the software is copied, for example, by the addition of 7 each time. To obtain the activation number, the user must provide to the software supplier the serial numbers of the hardware and the software and the number of copies which have been made. The software supplier may have a "hot line" phone to permit the user to obtain the activation code.

The Logan method relies on hiding a software code in a hidden location within the software. Thus, this code undesirably is within access by the user to allow formulation of an activation code (without making a telephone call to legitimately obtain it) and subsequent installation of the software to be achievable if the hidden code is located and the process then reverse-engineered. The hardware code which is used to generate the activation code is actually whatever number is inputted by the user and given to the supplier by the user. Since the software is not required to confirm that the activation number is based on the serial number of the specific computer to be authorized, the activation code which is supplied will allow installation of the software on any computer.

Various other techniques for preventing unauthorized software use are disclosed in U.S. patents 4,829,296; 4,866,769; 4,593,353; 4,683,553; 4,796,220; 5,263,157; 5,287,408; 5,311,591; and 5,293,422.

The above software-protection techniques can either be circumvented or rely on codes that can be broken or are so inconvenient to the customer that the competitive position of the software publisher suffers.

It is accordingly an object of the present invention to provide for authorization of a particular computer system or network by means of a technique for uniquely identifying the computer system or network so that the identification doesn't change over time whereby the software does not become unavailable on the authorized computer system or network.

It is a further object of the present invention to provide an easy and convenient means for activation of a software package on a particular computer system or network by a customer.

It is yet another object of the present invention to provide activation of a software package on a particular computer or computer network by means which cannot be discovered and copied by even computer-aided reverse engineering.

It is a still further object of the present invention to provide activation of a software package on a particular computer inexpensively and quickly.

In accordance with the present invention, a method is provided for preventing use of software on an unauthorized computer wherein the software is programmed to generate and output to the user of a computer a first or validation number derived from one or more of the following computer characteristics: serial number of the hard disk, the BIOS data from ROM, the number of sectors per track of the hard disk, the number of heads of the hard disk, and the number of cylinders of the hard disk. A second or activation number derived from the first number is encrypted by operation of a second computer at a remote location inaccessible to the user for input to the user's computer to allow use of the software on the user's computer. The second number includes one or more randomly generated digits which, when a predetermined mathematical operation is performed thereon and on at least one of the digits of the validation number, yields a derived balance number. This derived balance number is used in the user's computer to encrypt a thumbprint of the computer characteristic including a preselected signature and a productprint. When the computer program is to be executed, the software decodes the thumbprint and productprint using a predetermined balance number. If the derived balance number is equal to the predetermined balance number, the program will execute. Otherwise, it will not execute.

The above and other objects, features, and advantages of the present invention will be apparent in the following detailed description of the preferred embodiments thereof taken in conjunction with the accompanying drawings wherein the same reference numerals denote the same or similar parts throughout the several views.

Brief Description of the Drawings

Fig. 1 is a perspective view of a personal computer and a floppy disk within which is stored a computer program wherein the computer is to be authorized for use of the program therein in accordance with the present invention.

Fig. 2 is a generally diagrammatic view illustrating the hard disk drive therefor.

Fig. 3 is a generally diagrammatic view illustrating the software activation process which embodies the present invention.

Fig. 4 is a block diagram of the process.

Fig. 5 is a flow diagram therefor.

Fig. 6 is a flow diagram of a process for generating a validation number therefor.

Fig. 7 is a flow diagram of a process for generating from the validation number an activation number.

Fig. 8 is a flow diagram for execution of the software.

Fig. 9 is a flow diagram of the entering of the activation number by the user.

Fig. 10 is a flow diagram of generation of a thumbprint in the computer.

Fig. 11 is a diagrammatic view illustrating the thumbprint format in the computer.

Fig. 12 is a view similar to that of Fig. 11 illustrating the productprint format in the computer.

Fig. 13 is a diagrammatic view of the thumbprint/productprint areas illustrating scrambling of the thumbprint.

Fig. 14 is an enlarged view of the thumbprint area of Fig. 13.

Fig. 15 is a view similar to that of Fig. 3 illustrating an alternative embodiment to the present invention.

Fig. 16 is a flow diagram similar to that of Fig. 6 illustrating an alternative method of generating the validation number.

Fig. 17 is a flow diagram similar to that of Fig. 7 illustrating an alternative method of generating the activation number.

Detailed Description of the Preferred Embodiments

Referring to the drawings, there is shown in Fig. 1 a typical personal computer 10 of a type well known in the art and commercially available from a variety of manufacturers, for example, IBM Corporation. The personal computer 10 includes a standard keyboard

12, a standard cathode ray tube (CRT) or screen 14, and a pair of floppy disk drives 16. The keyboard 12 is employed to facilitate communication between an individual user, illustrated at 40 in Fig. 3, and the computer 10 in a manner which is generally well known in the computer art. The CRT 14 also functions in a manner well known in the computer art for displaying information inputted through the keyboard 12 as well as information outputted by the inner workings of the computer 10. The disk drives 16 are employed in a manner well known in the computer art for receiving one or more floppy disks to facilitate the loading or entry of computer software or programs stored within a floppy disk into the computer 10. A typical floppy disk 18 is illustrated in Fig. 1. As used herein, the terms, "program," "computer program," "software" and "software program" are interchangeably used to mean a series of instructions which are used to control the operation of computer hardware or other computer-based or process-based hardware. The reference numeral 18 will be used herein to refer interchangeably to the floppy disk as well as the program contained thereon.

While in the present description of a preferred embodiment of the invention, a personal computer 10 is shown and described, it will be appreciated by those skilled in the art that the present invention may be employed in conjunction with any other type of computer, including standard computers such as a microcomputer, a mini-computer, a main-frame computer, a computer network, and/or special purpose computers. In addition, the present invention may be employed in connection with any other type of computer or processor-based hardware such as computer or processor controlled machinery or equipment. By "computer network" is meant a plurality of computers which communicate via a client server, peer-to-peer, or the like.

Likewise, while in connection with the description of the presently preferred embodiment, the computer program or software is illustrated as being stored within a floppy disk 18, it will be appreciated by those skilled in the art that the program or software could alternatively be stored in any other type of storage medium, for example, a different magnetic medium, such as a CD-ROM drive, a hard disk drive, magnetic tape, etc.; a semiconductor based storage medium, such as a random access memory (RAM), a read only memory (ROM), a programmable read only memory (PROM), etc.; or a nontraditional storage medium, such as a digital audio or video tape or disk or network of storage devices. Accordingly, it should be clearly understood that the present invention is not limited to the particular computer hardware 10 or storage medium 18 used to illustrate the preferred embodiment of the invention.

Referring to Fig. 2, there is illustrated at 20 a fixed or hard disk drive for computer 10 which includes a multiplicity of platters 22 rotatable about a hub 24. Each platter 22 contains a plurality of concentric circular tracks 26 each containing a plurality of sectors 28 used for storage of digital informaticn. Although there are physically fewer sectors 28 in the tracks 26 closer to the hub 24, the hard drive controller, illustrated at 32, manages the space so that, as seen by the computer 10, there are on average typically 17 sectors 28 per track 26. Each platter 22 is two-sided and has on each side a read/write head 30 which magnetically stores onto and reads digital information from the platter 22.

For a track 26, there is a similarly situated track on the opposite side of its platter 22 and on each of the sides of the other platters, which multiplicity of tracks together is defined herein as a cylinder, illustrated at 34. A cylinder 34 is a logical ordering so that the controller 32 can simultaneously write to both sides of each of a multiplicity of platters 22.

Referring to Figs. 3 and 4, in accordance with the present invention, when a purchaser 40 of a publisher's software package 18 wishes to use the software on the computer 10, the software requires that it first be authorized. The software 18 is embedded with a program which prevents use of the software (or copies thereof) on a computer unless authorization is obtained for use on the particular computer. In a computer network, a maximum number of concurrent users may be authorized for use of the software, as described hereinafter.

The program 18 encrypts from one or more computer characteristics, as indicated at 42, a first or validation number, as indicated at 44, which appears on the computer screen along with instructions for obtaining a second or activation number for inputting to the computer 10, as indicated at 46, for executing the software 18, as indicated at 48.

In order that there be minimal inconvenience to the user 40, he or she is preferably instructed to call an "800" or the like phone number at an activation center, illustrated at 61, at another location (remote location) which is provided as a service to the publisher of the software 18. Thus, phones 52 and 54 respectively are used to orally communicate the validation number (and other information to be described hereinafter) over phone line 56 to the activation center operator 50 who then inputs via keyboard 58 the validation number to a second computer 60, which may be similar to computer 10 or another suitable conventional computer. This number is then used by the program 63 in computer 60 to generate and encrypt an activation number, as indicated at 62. The reference numeral 63 refers to a hard disk drive in computer 60 as well as a program stored thereon. The activation number is generated to be related to the validation number so that a number, herein called a "derived balance number," may be derived therefrom, as hereinafter discussed. The activation number is then provided by the operator 50 to the user 40 over phone line 56, who then inputs it to computer 10 by means of keyboard 12. The software program 18 then utilizes the validation and activation numbers, as indicated at 64, to obtain the derived balance number. If the validation and activation numbers have been correctly generated and inputted to the user's computer, the derived balance number will be equal to a predetermined balance number. This derived balance number is then used to encrypt a thumbprint of the computer characteristics including a preselected signature (TP) and a productprint (PP), as indicated at 65. For the software to be executed, as indicated at 124, the program is loaded to the hard disk 20, as indicated at 120, and the thumbprint and productprint are decrypted using the predetermined balance number, as indicated at 67. It is envisioned that, with CD-ROM or some other medium, the software program may not be loaded to the hard disk. If the preselected signature is retrieved, the program 18 proceeds with execution of the software, as indicated at 48.

As used herein and in the claims, a "predetermined balance number" is a number which is embedded in the software 18 or otherwise provided to decrypt the preselected signature, and a "derived balance number" is a number which is derived mathematically from the validation and activation numbers for encrypting the signature. As used herein and in the claims, a "signature" or "preselected signature" is information in the form of a preselected set of digits or characters which the software 18 is programmed to recognize or locate upon use of a decryption process using the predetermined balance number in order that the software be authorized for use. Therefore, if the derived balance number is the same as the predetermined balance number, the signature will be correctly encrypted and can as a result be decrypted by the predetermined balance number to yield the preselected signature whereby the program may be executed. Otherwise, the preselected signature cannot be found and the program will not execute.

Referring to Fig. 15, there is illustrated an alternative embodiment wherein person-to-person phone communication over telephone line 56 is replaced by modem-to-modem communication. Thus, modems 53 and 55 may be provided for computers 10 and 60 respectively for transmitting and receiving the needed information.

Fig. 5 illustrates in greater detail at 65 the process for activation of the software 18. As illustrated therein, the user 40 begins the process by inserting the diskette or CD-ROM or the like containing the software 18 in the respective drive 16. Alternatively, the user may have previously down-loaded (by modem) an embedded software package from a computer bulletin board service or other electronic distribution service. In this case, the software will be residing on the hard disk drive, awaiting activation. In all cases, the user selects the "activate" or "install" option. The software application code then checks for previous activation of this software package 18 on this particular computer system 10, i.e., is there a valid thumbprint/productprint (TP/PP) for this product. If "yes," the program may proceed with installation or re-installation of the software 18 without a call to the activation center. If "no," a first screen appears which greets the user 40 in the publisher's name and prompts the user to exit or to proceed with software activation.

If the user elects to proceed with software activation, the application code reads the system characteristics, which will be discussed hereinafter, and a second screen appears showing the publisher's name, product and version, customer identification, and product identification. The user is then requested to enter the publisher's product serial number after which it is validated for transcription errors. The user is requested to have basic demographic information available before making a "1-800," "1-900," "DDD," or the like telephone call to the activation center 61 and is then requested to call the activation center 61.

At the activation center 61, the operator 50 requests the customer's identification number, the product identification number, and published product serial number and displays the customer screen. The operator then receives and enters this information in the activation center computer 60. The last two digits of each of these numbers are check digits, determined in accordance with principles commonly known in the art to which this invention pertains, by means of which the program 63 checks whether the numbers are valid numbers. The operator may then receive and enter demographic information from a new customer or updated demographic information from an existing customer. The program 18 then proceeds to generate from the system characteristics a validation number which then appears on the screen. The operator 50 then requests and enters the validation number in the activation center computer 60, and the program 63 in the activation center computer proceeds to generate an activation number, as described hereinafter. This activation number is then relayed by phone from the operator 50 to the user 40, who then enters the information in computer 10. As previously discussed, this information may alternatively be transmitted back and forth by modem-to-modem communication. After deriving the balance number, the program 18 then "writes" the product identification, the computer characteristics, and the preselected signature in the form of a thumbprint/productprint (TP/PP) encrypted by the derived balance number, as described hereinafter, to the hard disk drive 20. If the activation number is not a correct number to generate a derived balance number which is the same as the predetermined balance number, then the TP/PP will be encrypted and written using a different number, and the preselected signature will not be found when subsequently applying the decryption process using the predetermined balance number. As a result, future efforts to execute previously authorized computer programs on this computer system will be unsuccessful. The screen will then prompt the user to proceed with installation of the computer program or to exit. If the user selects "proceed", the publisher package installation proceeds, and, when complete, the user system returns to the operating system prompt.

If the system characteristics on which the validation number is based have a tendency to change over time or are not sufficiently unique, as are the characteristics disclosed in the Durst, Jr. et al patent, then authorization of a computer may be unreliable in that the authorization may be lost if the characteristics change or the software may not reliably be prevented from use on an unauthorized computer system. Thus, the characteristics of the computer system on which the validation number is based are chosen to be unique and unchanging so that subsequent program execution on the same computer system is seamless yet attempts to execute the program on a different computer system will result reliably in the program being prevented from executing without a further authorization from the activation center. A suitable set of computer characteristics (32 bytes), which are available on standard industry hardware by accessing various interrupts and direct read functions in "C" language, using principles commonly known to those of ordinary skill in the art to which this invention pertains, are the serial number of the hard disk 20 (20 bytes), the BIOS data from ROM (read only memory), i.e., the date (MM/DD/YY) the system board for computer 10 was manufactured (8 bytes), and disk information consisting of the number of sectors 28 per track 26 (1 byte), the number of heads 30 (1 byte), and the number of cylinders 34 (2 bytes). It should be understood that the set of characteristics may be less than the above as long as the desired uniqueness is obtained. For example, the serial number of the hard disk 20, which includes a unique manufacturer identification number, may be sufficient. For another example, the combination of the BIOS data and the hard disk information may be sufficient.

Hereinafter, specific processes for generation of the validation and activation numbers, along with examples, will be provided. It should be understood that various variations may be made in these specific processes. Thus, neither the specific process steps nor the examples should be viewed as limiting the present invention but are instead to be taken as exemplary thereof.

Referring to Fig. 6, after the program 18 retrieves internal characteristic information, as indicated at 70, these 32 bytes of information are reduced to 4 internal random bytes (for example, A.!Ø), as indicated at 72, by the conventional technique of a recursive modulus 256 check-sum procedure, a technique commonly known to those of ordinary skill in the art to which this invention pertains. Each of the four bytes correspond to numbers between 0 and 255, for example, 61, 128, 85, 40. The reduction in the number of bytes is primarily to reduce the volume of information to be transmitted over the phone by the user and operator. However, with modem-to-modem communication, as previously discussed relative to Fig. 15, it may be unnecessary to reduce the 32 bytes to 4 since convenience of the user and operation would no longer be a consideration.

As indicated at 74, 5 check digits are calculated from these four bytes by a conventional weighted technique wherein the summation of the products of the bytes and weighted numbers, using the weighting 2, 3, 4, and 5 respectively, is divided by 10, and the remainder is the first check digit D_1. Thus, $D_1=2$ as follows:

$$[5(61)+4(128)+3(85)+2(40)]/10$$
$$= 115, \text{ remainder } 2$$

Check digit D_1 is appended to the four bytes, i.e., 61, 128, 85, 40, 2, for calculation of check digit D_2, and the summation of the products of the bytes (with D_1) and numbers 2, 3, 4, 5, and 6 (shifted to the right) respectively is divided again by 10, and the remainder is the second check digit D_2. Thus, $D_2=0$ as follows:

$$[6(61)+5(128)+4(85)+3(40)+2(2)]/10$$
$$= 147, \text{ remainder } 0$$

The remaining check digits D_3, D_4, and D_5 may be calculated similarly with "shifting to the right" occurring for each check digit. As illustrated at 76, these check digits are placed in an intermediate storage buffer to await the generation of 5 random digits, as hereinafter discussed.

Meanwhile, as indicated at 78, the program 18 generates the 5 random digits R_1 to R_5. As indicated at 80, these random digits R_1 to R_5 are added respectively to the check digits D_1 to D_5 (and any resulting digit in the 10s column dropped) to obtain a set of digits C_1 to C_5. For example, assuming D_1 to D_5 = 3, 5, 1, 9, 2, and R_1 to R_5 = 8, 3, 6, 2, 5, C_1 to C_5 are calculated as follows:

$$\begin{array}{cccccl} 3 & 5 & 1 & 9 & 2 & D_1 \text{ to } D_5 \\ \underline{8} & \underline{3} & \underline{6} & \underline{2} & \underline{5} & R_1 \text{ to } R_5 \\ 1 & 8 & 7 & 1 & 7 & C_1 \text{ to } C_5 \end{array}$$

As indicated at 82, the digits C_1 to C_5 and the random digits R_1 to R_5 are assembled as $R_1 \ldots R_5$, $C_1 \ldots C_5$, i.e.,

8 3 6 2 5 1 8 7 1 7

As indicated at 84, two check digits C_6 and C_7 are calculated similarly as discussed for check digits D_1 to D_5. Thus, C_6 is calculated by summing the products of the 10 digits and 2, 3, 4, 5, 6, 7, 8, 9, 2, 3 respectively and dividing by 10, the remainder being C_6 which, in this example, is 6, as follows:

$$[2(7)+3(1)+4(7)+5(8)+6(1)+7(5)+8(2)+9(6)+2(3)+3(8)]/10 = 22, \text{ remainder } 6.$$

Using the resulting 11 digit number and shifting to the right, check digit C_7 = 8, as follows:

$$[2(6)+3(7)+4(1)+5(7)+6(8)+7(1)+8(5)+9(2)+2(6)+3(3)+4(8)]/10 = 23, \text{ remainder } 8.$$

As indicated at 86, random numbers R_1 to R_5, digits C_1 to C_5, and the check digits C_6 and C_7 are assembled into the validation number $R_1 \ldots R_5$, $C_1 \ldots C_5$, C_6, C_7 which, in the example, is:

Validation no: 8 3 6 2 5 1 8 7 1 7 6 8

Thus, the resulting pseudo-random validation number generated by the program 18 in the user's computer 10 comprises digits which are meaningless to the user and have no meaning relative to the computer characteristics, except that the computer characteristics can be derived therefrom by means of a program which traces backwardly the validation code to the original 32 bytes. Since the process is pseudo-random, the derivation of such a program by a hacker is not envisioned. By re-calculation of check digits C_6 and C_7, the activation computer 60 can confirm that the validation number provided by the user 40 is a correct and not a fabricated or incorrectly given validation number.

Referring to Fig. 16, there is illustrated an alternative method of generating the validation number which allows the authenticity of the customer and product identification and the product serial no. to be checked for relational correctness and whether the information given over the phone corresponds to what is entered in the computer 10. Often, the product identification and product serial numbers are within a range of numbers, permitting a further check on their correctness.

As indicated at 200, the customer and product identification numbers, the publisher's serial number, and the preliminary validation number (including check digits) are first assembled into a number (customer

ID....C_7), the preliminary validation number in this embodiment being defined to be the same as the 12-digit validation number previously discussed. This assembled number is then used to generate from all of the bytes thereof two check digits C_8 and C_9, as indicated at 202, in a manner as previously discussed for generation of check digits. As indicated at 204, the resulting number with these check digits appended (customer ID....C_9) is then summed. Two more check digits C_{10} and C_{11} are then generated based on the sum, as indicated at 206, again using similar principles for check digit generation. The check digits C_8, C_9, C_{10}, and C_{11} are appended to the preliminary validation number to obtain a final validation number (R_1...R_5, C_1....C_{11}), as indicated at 208.

At the activation center, the check digits C_8 to C_{11} will be used to determine if the information given by the user checks, i.e., the activation center will double-check to determine if the user really gave the correct information.

Unless otherwise noted, the term "validation number" will refer in this specification to the 12-digit validation number but may refer in the claims to either validation number or another suitable validation number.

Referring to Fig. 7, there is indicated the process of generation of the nine digit activation number A_1 to A_9 by program 63 in the remote activation computer 60. As indicated at 90, after the validation number is inputted, the check digits C_6 and C_7 are re-calculated and compared with the corresponding digits in the validation number as supplied over the phone by the user to confirm the validation number as a correct one which has not been fabricated or incorrectly given by the user.

As indicated at 92, the sum of the digits of the validation number is calculated, this sum being a number which is defined herein as "Balance 1." Thus, in this example,

Balance 1 = 8+3+6+2+5+1+8+7+1+7+6+8 = 62

It should however be understood that Balance 1 may be obtained from the validation number by any other suitable mathematical process.

As indicated at 94, digits A_2, A_6, and A_7 are calculated from the validation number as follows. A_2 is set equal to the unit's value of balance 1, and A_6 is set equal to the ten's value thereof. Thus, in the example, A_2 = 2 and A_6 = 6. The digits R_1... R_5, C_1... C_5 are multiplied respectively by 1, 2, 4, 8, 16, 32, 64, 128, 256, 512 (hexadecimal weighting, i.e., the digits being multiplied respectively by a set of numbers with each being double the preceding number, beginning with 1), and the summation of the products is divided by 10, the remainder being A_7. Thus, A_7 = 4, calculated as follows:

[1(8)+2(3)+4(6)+8(2)+16(5)+32(1)+64(8)+128(7)
+256(1)+512(7)]/10 = 541, remainder 4

As indicated at 96, three random digits a, b, and c are generated by the program 63. A number d is calculated as a(b)+c, as indicated at 98. As indicated at 100, if d is greater than or equal to a predetermined balance number, d is subtracted from Balance 1, giving e. Otherwise, d is added to Balance 1, giving e. As indicated at 102, if the result e is not equal to the predetermined balance number, a new set of 3 random digits is generated and steps 96, 98, and 100 re-applied until a set of 3 digits a, b, and c is randomly selected such that e is equal to the predetermined balance number. The number d for those three digits (wherein, e = the predetermined balance number) is defined herein as "Balance 2," and those three digits a, b, and c are set equal to A_3, A_1, and A_5 respectively, as indicated at 104. It should be understood that Balance 2 may be obtained from the three (or other suitable number) of randomly-generated digits by any other suitable mathematical process. Thus, Balance 2, in this example, is equal to Balance 1 less the predetermined balance number, i.e., Balance 2 = 62 - 5 = 57.

It is preferred that the predetermined balance number be a prime number such as, in the example, 5, since a factorable number is weak mathematically so that the code may be more easily cracked. More preferably, the prime number is a higher number such as a 2, 3, or 4 digit prime number since more digits of information are involved, making any effort to determine the predetermined balance number even more difficult.

With e = 5, a random set of values for a, b, and c may be 7, 8, and 1 respectively whereby A_1, A_3, and A_5 are 8, 7, and 1 respectively. This is because
Balance 2 = 7(8) + 1 = 57, and
e = Balance 1 - Balance 2 = 62 - 57 = 5.

As indicated at 106, a determination is made whether Balance 1 is greater than or equal to the predetermined balance number e. If Balance 1 is less than the predetermined balance number, 5, then A_4 is set to a random number of 0 to 4, as indicated at 108. If Balance 1 is greater than or equal to the predetermined balance number, 5, as it is in the example, then A_4 is set to a random number of 5 to 9. For example, A_4 may be set randomly to 6.

As indicated at 112, the digits A_1 to A_7 are then assembled, as follows:

8 2 7 6 1 6 4

Similarly as check digits C_6 and C_7 were calculated for the validation number, check digits A_8 and

A_9 are calculated for the activation number, as indicated at 114, except the multipliers of A_1 to A_7 and then A_1 to A_8 begin with different digits, i.e., 7 and 8 respectively. Thus, in the example, A_8 = 6 as follows:

[7(4)+8(6)+9(1)+2(6)+3(7)+4(2)+5(8)]/10
= 16, remainder 6

A_9=6 as follows:

[8(6)+9(4)+2(6)+3(1)+4(6)+5(7)+6(2)+7(8)]/10
= 22, remainder 6

As illustrated at 116, the activation number A_1 to A_9 is assembled and displayed on the screen to the operator 50, as follows:

8 2 7 6 1 6 4 6 6

This number is delivered over the phone, by modem, or otherwise to the user for inputting to computer 10.

Referring to Fig. 17, there is illustrated an alternative method of generating the activation number, which allows a greater check on the authenticity of the digits thereof. After assembly of the preliminary activation number, as illustrated in Fig. 7 and indicated at 116, this assembled number is then used to generate from all of the bytes thereof two check digits A_{10} and A_{11}, as indicated at 250, in a manner as previously discussed for generation of check digits. As indicated at 252, the resulting number with these check digits appended (A_1....A_{11}) is then summed. Two more check digits A_{12} and A_{13} are then generated based on the sum, as indicated at 254, again using similar principles for check digit generation. The check digits A_{10}, A_{11}, A_{12}, and A_{13} are appended to the preliminary activation number to obtain a final activation number, as indicated at 256. When the final activation number is inputted to the user's computer, the program will utilize these additional check digits to determine if the activation number is a correctly generated number.

Unless otherwise noted, the term "activation number" will refer in this specification to the nine-digit activation number but may refer in the claims to either activation number or another suitable activation number.

As with the validation number, the resulting pseudo-random activation number generated by the program 63 in the remote computer 60 comprises digits which are meaningless to the user and have no meaning relative to the validation number, which is also meaningless to the user. If, however, the user were to successfully generate an activation number which would cause the TP/PP to be encrypted and written, the TP/PP cannot thereafter be decrypted to retrieve the preselected signature for execution of the software unless the random digits were also selected to give a derived balance number which is the same as the predetermined balance number.

The activation number is given over the phone, modem, or the like to the user 40 and inputted to the computer 10 being authorized. The program 18 then generates a derived balance number and causes the customer and product identification, computer system unique characteristics, and the preselected signature to be written on the hard disk drive 20 as the thumbprint/productprint (TP/PP), encrypted by use of the derived balance number, as described hereinafter, preferably in several locations to facilitate data integrity/recovery across all operating systems, i.e., DOS, Windows, OS/2, and the like: (1) one or more locations in the root of the hard disk drive 20, i.e. a non-hidden file in the directory, (2) track 0 of the first cylinder 34, i.e., a hidden file, and (3) several (perhaps 3) locations on the diagnostic cylinder, i.e. hidden files. By "hidden files" is meant that there is no directory in the system which indicates their existence. The information is also written to several different locations as a back-up, i.e., in case it gets inadvertently deleted at one or more locations. There are situations where it may be desirable to recover the information. For example, during unloading and reloading, the "non-hidden file" and its root directory will be recovered, i.e., rewritten. Therefore, it is desirable that the information be written to the root directory even though it is not a hidden file.

Referring to Fig. 8, for execution of the software package 18, the user "runs" the software, as indicated at 120, and the "executable" code portion thereof checks for whether a valid thumbprint/productprint (TP/PP) exists on the hard disk drive 20, as indicated at 122. If a valid TP/PP has been written to the hard disk drive 20, the software executes, as indicated at 124. A wrapper in each software package may have several "enabling" function calls to the embedded, encrypted "code." However, if a valid TP/PP does not exist, then the software causes the computer screen to display a message prompting the user to insert the activation/installation diskette, CD ROM, or the like medium in order to activate, as indicated at 126. If the medium is inserted, the activation process begins, as indicated at 128. Otherwise, the program terminates and a return to the Operating System occurs.

Referring to Fig. 9, after the activation number A_1 to A_9 is inputted to the computer 10 by the user 40, as indicated at 130, check digits A^1_8 and A^1_9 are calculated (in the same way check digits A_8 and A_9 were calculated) and compared with digits A_8 and A_9, as indicated at 132 and 134 respectively. If $A_8 = A^1_8$ and

$A_9 = A^1_9$, then the program proceeds to a calculation of A^1_2, A^1_6, and A^1_7 (in the same way A_2, A_6, and A_7 were calculated) and a comparison made with A_2, A_6, and A_7, as indicated at 136 and 138 respectively. If $A_2 = A^1_2$, $A_6 = A^1_6$, and $A_7 = A^1_7$, then A_1, A_3, and A_5 are extracted from the activation number, Balance 1 is set equal to the sum of the digits of the validation number, and Balance 2 is set equal to $A_3(A_1) + A_5$, as indicated at 140, 142, and 144 respectively. A_4 is abstracted from the validation number. If A_4 is greater than or equal to 5, then the derived balance number is equal to Balance 1 less Balance 2, as indicated at 146. Otherwise, the derived balance number is equal to the sum of Balance 1 and Balance 2, as indicated at 148.

Referring to Figs. 10 to 14, the derived balance number is used to encrypt and write to the hard disk drive 20 the thumbprint in a thumbprint format, indicated at 209 in Fig. 11, and the productprint (containing the publisher's product identification number in a productprint format), illustrated at 221 in Fig. 12, contained within a cluster of perhaps 4 sectors 28 (2048 bytes), as seen in Fig. 13. The thumbprint 150 is contained within one of the sectors (512 bytes).

As indicated at 160 in Fig. 10, the program first checks for whether a thumbprint 150 exists. If it does, it is then updated for a new productprint, as indicated at 161, and a random number generator is run to determine randomly a "pointer" start position, as indicated at 163. If it doesn't, a thumbprint 150 must be generated. This is done by running a unique random number generator for the "pointer" portion 154 (right side 256 bytes) of the thumbprint area 150, as indicated at 162, running a non-unique random number generator for the "data" portion (left side 256 bytes) of the thumbprint area 150 and the 3 sectors for productprints, as indicated at 164, and running a random number generator to determine randomly a "pointer" start position, as indicated at 166.

Referring to Fig. 11, the thumbprint is assembled in the area 150 in a format, indicated at 209, of perhaps 35 bytes including (1) the authorizer's signature (16 bytes), illustrated at 226, (2) the customer identification number (4 bytes), illustrated at 210, (3) the number of products for this customer number (2 bytes, based on how many productprints have been written), illustrated at 212, (4) a productprint encryption key (2 bytes, a random number used to encrypt the productprint by a suitable conventional process), illustrated at 214, (5) the 4-byte internal machine characteristic data, illustrated at 216, (6) four pointers (1 byte each), illustrated at 218, used for recovery of the TP/PP in track zero since they identify 4 particular sectors previously allocated by the operating system therefor, and (6) a check sum (3 bytes), i.e., which is derived by the modulus 256 process as previously discussed, illustrated at 220.

Referring to Fig. 12, the productprint, encrypted by encryption key 214 and then XOR'd to reverse bits in accordance with principles commonly known to those of ordinary skill in the art to which this invention pertains, is assembled in the area 152 in a format, illustrated at 221, of perhaps 11 bytes including (1) product identification (2 bytes), illustrated at 222, (2) "try & buy" indicators (3 bytes), illustrated at 223, (3) network indicators (3 bytes), illustrated at 224, and (4) a check sum (3 bytes), illustrated at 225. The "try & buy" and "network" indicators 223 and 224 respectively will be discussed hereinafter. It should be understood that these indicators 223 and 224 are optional and need not be provided if the software package is not to have these features.

In order to determine if a software program is authorized or "maximum" users in a network are exceeded, the embedded software about to execute on a "node" is programmed to communicate back to the client server or peer-to-peer node to obtain authorization prior to executing. Character 6 in the "network" indicator 224 is a "type of network" designator, i.e., perhaps using the characters "N" for Novell, "B" for Banyan, "W" for Windows, "L" for Lantastic, and "A" for "not applicable." Characters 7 and 8 contain the maximum number of users allowed concurrently. If character 6 is "A" or another character indicating that the software contains no provision for network use, then characters 7 and 8 are random digits.

The thumbprint 209 also contains the preselected signature (16 bytes), illustrated at 226, which is a set of characters which are the same for each item of software 18. The preselected signature 226 may be determined randomly or in any other suitable way. For example, the signature may be generated by beginning with 28 and adding 91 (if the sum is greater than 255, then 255 is subtracted to get the number) until the 16 characters are generated. It is this signature which must be retrieved by the program 18 before execution of the software is permitted.

Referring to Figs. 13 and 14, the numbers generated in the pointer portion 154 of 256 bytes are random and unique, i.e., each number appears only once. Thus, in the example of Fig. 12, the first 6 bytes randomly contain unique numbers 56, 1, 14, 255, 48, and 4. A start-point byte is randomly selected, for example, at the third byte, indicated at 158, containing the number 14.

The 35 (or more) bytes of the thumbprint 209 are scrambled or randomly scattered in the "data" portion 156 as controlled by the "pointer" portion 154.

Thus, to assemble the thumbprint data, as indicated at 168, the start-point byte 158 determines the byte-position of the first byte of the thumbprint, i.e., byte number 14 in the "data" portion. The next pointer byte containing number 255 determines the byte-position of the second byte of the thumbprint, i.e., data portion byte number 255. The locations of the remaining thumbprint bytes are determined similarly, and the remaining or unused bytes in the "data" portion retain their randomly-generated numbers.

The program 18 proceeds to decompose the validation and activation numbers and obtain a derived balance number, as previously discussed relative to Fig. 9, which is used to encrypt the TP/PP, as illustrated at 174, by any suitable encryption method. For example, each encrypted byte may be used to encrypt the next byte in a ripple effect.

After the thumbprint is assembled or updated, the productprint data is assembled, as indicated at 170. New check sums are calculated and stored for the TP, PP, and cluster, as indicated at 172, followed by encrypting the PP with the randomly generated number in the TP (then XOR'd) and the TP/PP with the derived balance number, as indicated at 174. It is this encrypted TP/PP which is then written to the hard disk drive 20, as indicated at 176.

For execution of the software 18, the program effects decryption using the predetermined balance number. If the predetermined balance number is the same as the derived balance number (meaning that the validation and activation number set was correctly decomposable to yield a derived balance number which is equal to the predetermined balance number), then the preselected signature 226 as well as the remainder of the TP/PP will be retrieved. If the derived balance number is not the same as the predetermined balance number, the decryption will not yield the preselected signature 226, and the program 18 will not be executed. To throw a hacker further off guard, the application software is preferably decrypted and re-encrypted on the fly, i.e., as it is being run.

The predetermined balance number is suitably encrypted in object code which is given to the publisher to embed in the program 18, using principles commonly known to those of ordinary skill in the art to which this invention pertains. ^The publisher may not therefore know the balance number. A series of confusing processes are used, in accordance with principles commonly known to those of ordinary skill in the art to which this invention pertains, to deny access to the predetermined balance number to the user or a hacker.

Due to the pseudo-random nature of the validation and activation codes, different validation and activation numbers are obtained each time software is installed on the same computer, and these numbers disappear once the derived balance number is obtained.

In order to allow a potential customer to "try and buy", the software 18 is preferably programmed to allow activation then shut down (or provide a "nagging" message periodically) after a number of uses and/or number of days, as specified in the productprint 223. This would allow software to be, for example, placed on a bulletin board, downloaded, activated, and tried for a period of time, as specified by the publisher, and then be purchased during a brief telephone call to the activation center. The publisher selects the "nag" or "shutdown" version prior to package embedding.

Referring to Fig. 12, character 3 of the "try & buy" indicator 223 is an indication of whether or not the activated package has been purchased. If it has, character 3 may, for example, be a "P" for "purchased."

If it is in "try" mode, character 3 may be a character which indicates either "nag" (continue to operate when the specified number of units of time and "tries" have been used, but a reminder message on a regular basis) or "no nag" (shut down when the specified number of units of time or "tries" have been used). Character 3 also specifies the unit of time, i.e., seconds, minutes, hours, days, or months. Character 4 indicates the number of units of time allowed, and character 5 indicates the number of tries allowed. If character 3 contains a "P," then characters 4 and 5 are random characters. When the user purchases the software, character 3 is changed to the "buy" character.

Referring to Fig. 6, the following is an alternative method for implementing the "network" and "try & buy" features. Instead of listing a check digit, D_5 is selected to provide information relative to which of these features is to be implemented to be passed from the user 40 to the operator 50 (or between the respective computers) encoded within the digit D_5. The software 18 is programmed to check for the "network" and "try & buy" states and select a digit D_5 indicative thereof. The possible states for each feature are "yes" and "no." If the feature is not included as an option for the type of software, it is "inactive." For example, the digit D_5 may be selected as follows:

Network state	Try & Buy state	D_5
No	No	1
No	Yes	2
Yes	No	3
Yes	Yes	4
Inactive	Inactive	0

Thus, D_5, in this embodiment would not be a check digit but would be a digit selected to represent the "network" and "try & buy" states.

When the operator 50 enters this digit D_5 as part of the validation number, the computer 60 is programmed to update its information database 63 appropriately to reflect the user's "network" and/or "try & buy" implementation.

As previously discussed, the pseudo-random encrypting of the validation and activation numbers and the random scattering of the thumbprint/productprint information provides numbers which appear to be meaningless and would not be expected to be decoded by a hacker even by the sophisticated programs and techniques currently in use. The maintenance of the program for generating the activation number at the activation center is inaccessible to the user and maintains secure that information which is needed to decode the activation number. The process of the present invention therefore does not require hiding of codes within the software.

Even if the hacker were successful in determining the derivation of the validation number and the non-random activation number digits from the validation number, he or she may still be stumped by a failure to realize that the group of randomly generated digits A_3, A_1, A_5 must have balance relative to the digits of the validation number. Thus, a set of random digits A_3, A_1, A_5 will not allow correct installation of the software 18 unless the derived balance number is the same as the predetermined balance number. This "balance number" approach uniquely provides with the encrypted number generation a two-piece approach which is not a simple comparison which can be branched around by a hacker. Thus, even if the hacker is able to successfully break the code and generate an activation number which will cause an encrypted TP/PP to be written, his failure to select a set of random digits which will provide the predetermined balance number still prevents execution of the software since, upon decryption, the preselected signature cannot be located.

Thus, although a method for authorizing the use of a software package on a particular computer system is provided by the present invention so as to be virtually impossible, given the current state of the art, to decode or reverse engineer, the process is made convenient and easy for the software user, i.e., he or she need only make a phone call and follow some easy directions. With modem-to-modem communication, personal communication with an activation center operator is not even required. Further, the unchanging and unique nature of the computer characteristics on which authorization is based allow the authorization process to be reliable, i.e., an authorization on one machine does not include others, and the user can be assured that the authorization will not be lost just because the computer characteristics may have changed since authorization.

Although the invention has been described in detail herein, it should be understood that the invention can be embodied otherwise without departing from the principles thereof, and such other embodiments are meant to come within the scope of the present invention as defined in the appended claims.

What is claimed is:

1. A method for authorizing use of software on a computer comprising the steps of:

a. programming the software to encrypt and output to the user of a computer a first number derived from information received by the software from the computer of at least one characteristic of the computer providing an unchangeable and unique identification of the computer; and

b. Operating an other computer thereby encrypting a second number derived from the first number for input to the user's computer to allow use of the software on the user's computer only if a predetermined relationship exists between the first and second numbers.

2. A method according to claim 1 further comprising selecting the at least one computer characteristic from the group of computer characteristics consisting of the serial number of the disk drive, the BIOS data from ROM, the number of sectors per track of the hard disk, the number of heads of the hard disk, and the number of cylinders of the hard disk.

3. A method according to claim 2 comprising programming the software to encrypt the first number from information as to all of said group of computer characteristics.

4. A method according to claim 2 comprising programming the software to encrypt the first number from information as to the BIOS data from ROM, the number of sectors per track of the hard disk, the number of heads of the hard disk, and the number of cylinders of the hard disk.

5. A method according to claim 1 comprising securing information relative to the process for encrypting the second number from access thereto by the user.

6. A method according to claim 1 comprising encrypting the second number pseudo-randomly.

7. A method according to claim 1 further comprising programming the software to encrypt the first number from the serial number of the hard disk.

8. A method according to claim 1 further comprising programming the software to provide an option to the user of the computer for trial of the software

for a specified period such that a new authorization for use is required after the trial period is concluded.

9. A method according to claim 1 further comprising programming the software to allow the authorization for use of the software to cover a specified number of computers in a network.

10. A method for authorizing use on a computer of software which has been programmed to output to the user of a computer a first number derived from at least one characteristic of the computer, the method comprising operating an other computer thereby encrypting for inputting to the user's computer a second number derived from the first number and including at least one randomly generated digit which, when a predetermined mathematical operation is performed on said at least one randomly generated digit and on at least one of the digits of the first number, yields in the user's computer a derived balance number used to encrypt a preselected signature in the user's computer whereby the software may be used in the user's computer upon use of a predetermined balance number equal to the derived balance number to retrieve the preselected signature by decryption thereof.

11. A method according to claim 10 further comprising selecting the predetermined balance number to be a prime number.

12. A method according to claim 10 comprising securing information relative to the process for encrypting the second number from access thereto by the user.

13. A method according to claim 10 comprising encrypting the second number pseudo-randomly.

14. A method for authorizing use of software on a computer comprising the steps of:

a. programming the software to output to the user of a computer a first number derived from at least one characteristic of the computer; and

b. operating an other computer thereby encrypting for inputting to the user's computer a second number derived from the first number and including at least one randomly generated digit which, when a predetermined mathematical operation is performed on said at least one randomly generated digit and on at least one of the digits of the first number, yields in the user's computer a derived balance number used to encrypt a preselected signature whereby the software may be used in the user's computer upon use of a predetermined balance number equal to the derived balance number to retrieve the preselected signature by decryption thereof.

15. A method according to claim 14 further comprising programming the software to write information including said preselected signature encrypted by the derived balance number in the hard drive of the user's computer.

16. A method according to claim 15 further comprising programming the software to scatter the bytes of said encrypted information among randomly-generated bytes.

17. A method according to claim 14 further comprising programming the software to write information including said preselected signature encrypted by the derived balance number in the root of the hard disk of the user's computer, track 0 of the first cylinder of the user's computer, and at least one location on the diagnostic cylinder of the user's computer.

18. A method according to claim 14 further comprising selecting the at least one computer characteristic from the group of computer characteristics consisting of the serial number of the

hard disk, the BIOS data from ROM, the number of sectors per track of the hard disk, the number of heads of the hard disk, and the number of cylinders of the hard disk.

19. A method according to claim 14 further comprising selecting the at least one computer characteristic to provide an unchangeable and unique identification of the computer.

20. A method according to claim 14 further comprising programming the software to receive information relative to the at least one computer characteristic from the computer.

21. A method according to claim 14 further comprising programming the software to provide an option to the user of the computer for trial of the software for a specified period such that a new authorization for use is required after the trial period is concluded.

22. A method according to claim 14 further comprising programming the software to allow the authorization for use of the software to cover a specified number of computers in a network.

23. A method for authorizing use of software on a computer comprising the steps of:

a. programming the software to pseudo-randomly encrypt and output to the user of a computer a first number derived from information received by the software from the computer of at least one characteristic of the computer;

b. inserting the software in the computer and operating the computer to obtain the first number;

c. operating another computer thereby pseudo-randomly encrypting a second number derived from the first number and including at least one randomly-generated digit which, when a predetermined mathematical operation is performed on said at least one randomly generated digit and on at least one of the digits of the first number, yields a derived balance number;

d. inputting the second number to the user's computer;

e. operating the user's computer thereby obtaining the derived balance number;

f. operating the user's computer thereby assembling information including a preselected signature;

g. operating the user's computer thereby encrypting the assembled information using the derived balance number; and

h. operating the user's computer thereby writing the encrypted assembled information to the user's computer whereby the software may be used in the user's computer upon use of the predetermined balance number to retrieve the preselected signature by decryption thereof.

24. A method according to claim 23 further comprising connecting the user's computer and another computer by modems for communication between the user's computer and the another computer.

25. A method according to claim 23 further comprising selecting the predetermined balance number to be a prime number.

26. A method according to claim 23 further comprising operating the user's computer thereby writing the encrypted assembled information including the preselected signature to at least one location in the root of the hard disk drive, track 0 of the first cylinder, and at least one location on the diagnostic cylinder.

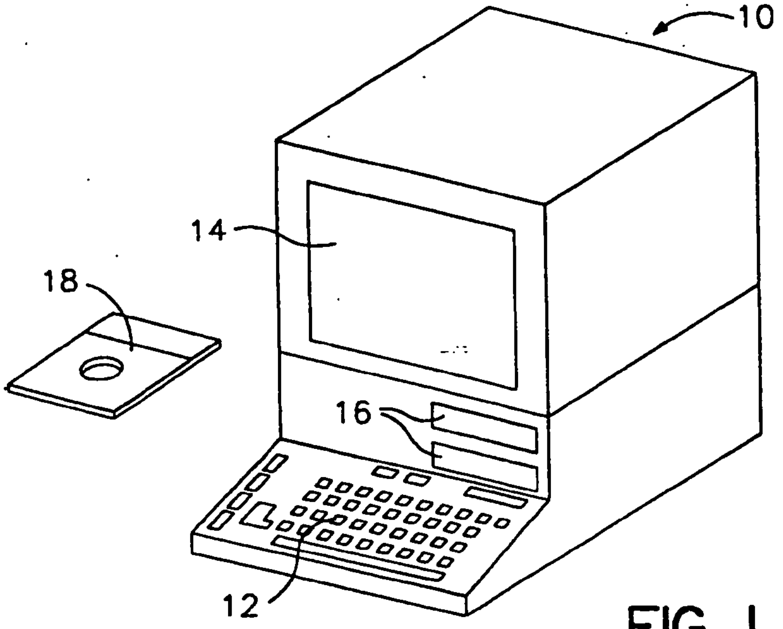


FIG. 1

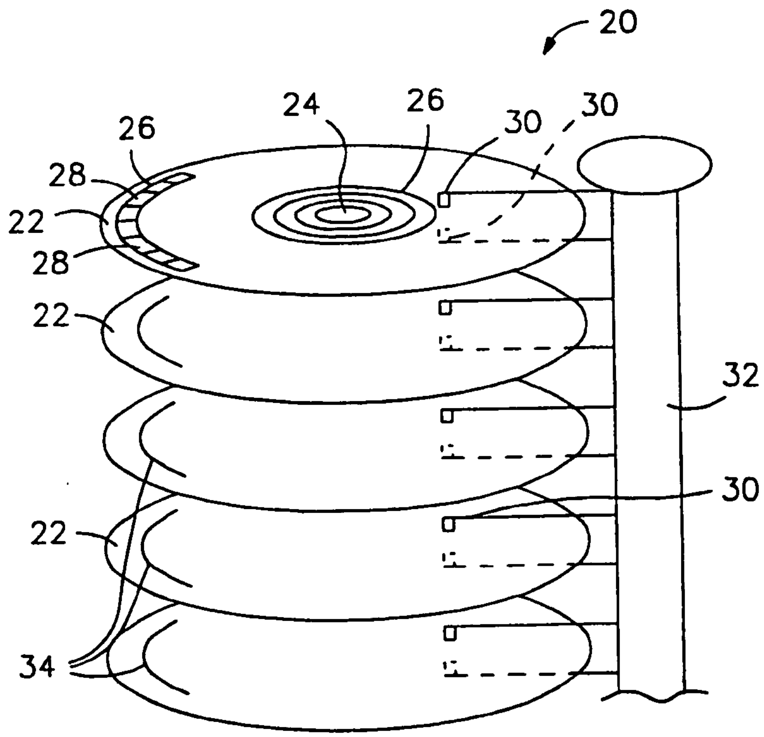


FIG. 2

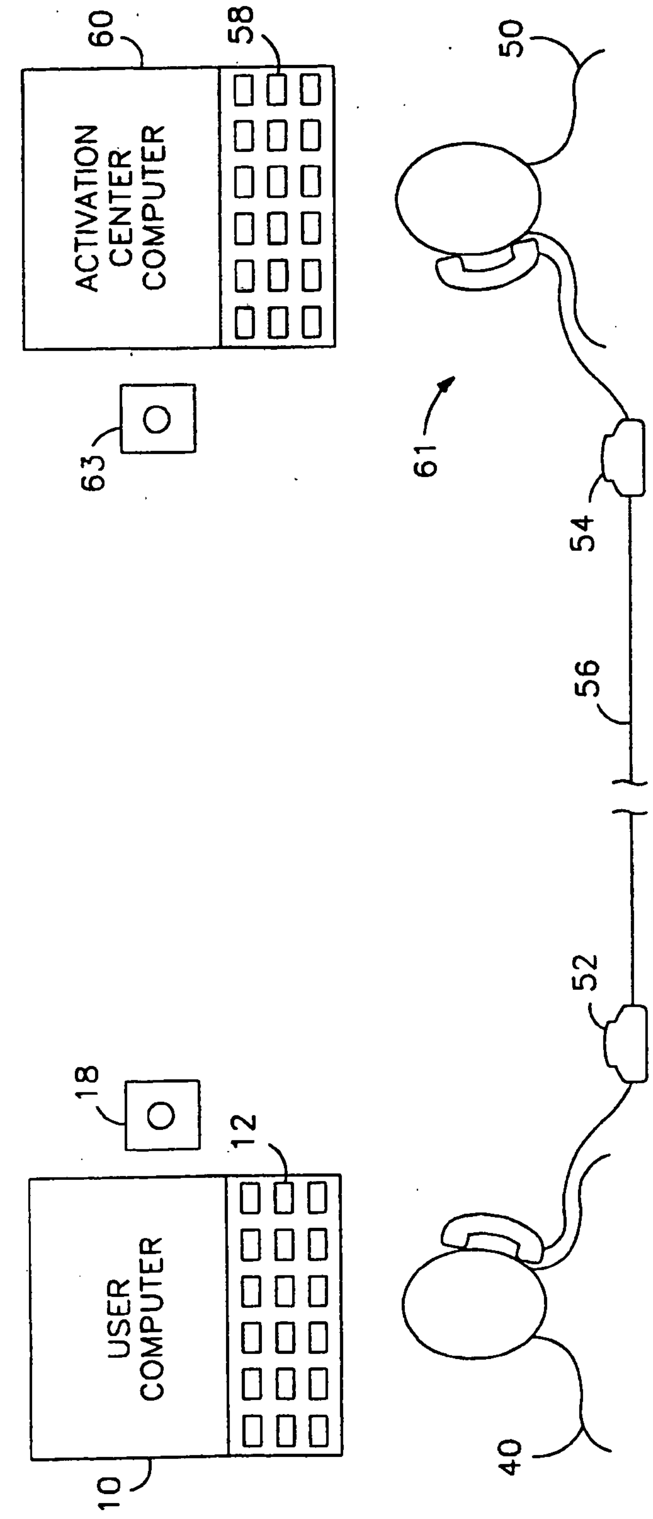

60
58
ACTIVATION CENTER COMPUTER
50
63
61
54
56
52
18
12
USER COMPUTER
40
10

FIG. 3

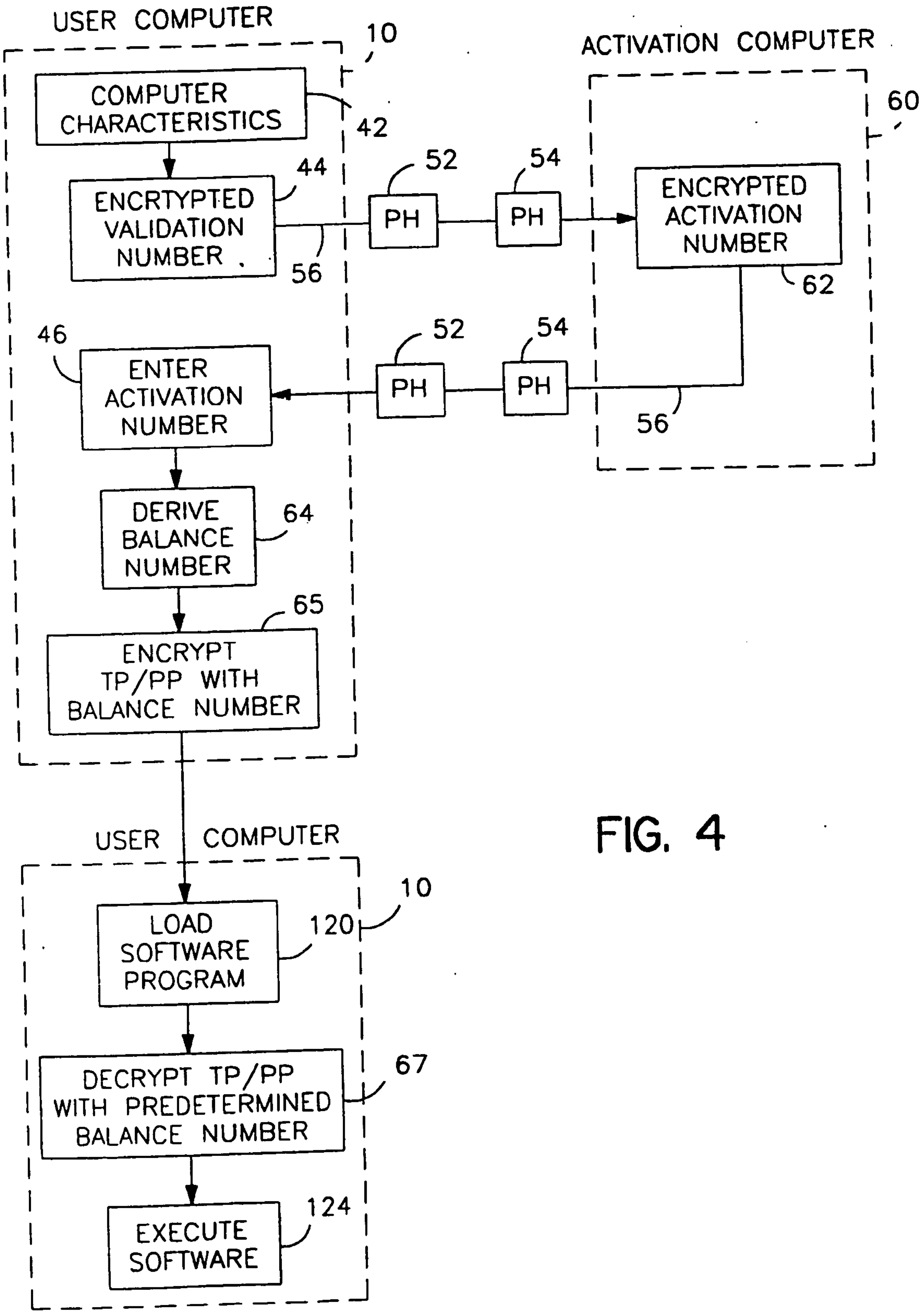

USER COMPUTER
10
COMPUTER CHARACTERISTICS
42
ENCRTYPTED VALIDATION NUMBER
44
56
52
PH
54
PH
ACTIVATION COMPUTER
60
ENCRYPTED ACTIVATION NUMBER
62
46
ENTER ACTIVATION NUMBER
52
PH
54
PH
56
DERIVE BALANCE NUMBER
64
65
ENCRYPT TP/PP WITH BALANCE NUMBER
USER COMPUTER
10
LOAD SOFTWARE PROGRAM
120
DECRYPT TP/PP WITH PREDETERMINED BALANCE NUMBER
67
EXECUTE SOFTWARE
124

FIG. 4

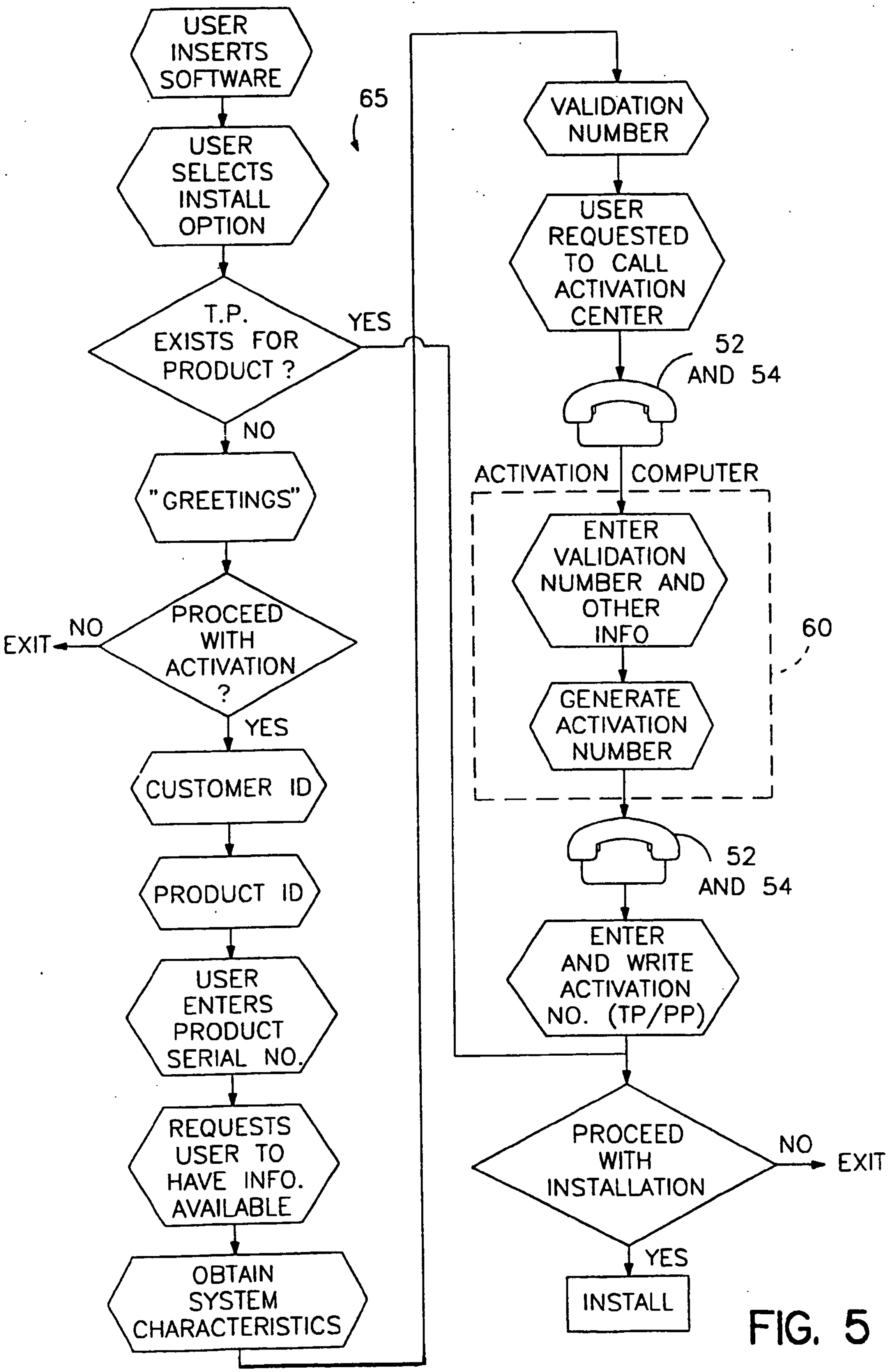

USER INSERTS SOFTWARE
65
USER SELECTS INSTALL OPTION
T.P. EXISTS FOR PRODUCT ?
YES
NO
"GREETINGS"
PROCEED WITH ACTIVATION ?
NO
EXIT
YES
CUSTOMER ID
PRODUCT ID
USER ENTERS PRODUCT SERIAL NO.
REQUESTS USER TO HAVE INFO. AVAILABLE
OBTAIN SYSTEM CHARACTERISTICS
VALIDATION NUMBER
USER REQUESTED TO CALL ACTIVATION CENTER
52 AND 54
ACTIVATION COMPUTER
ENTER VALIDATION NUMBER AND OTHER INFO
GENERATE ACTIVATION NUMBER
60
52 AND 54
ENTER AND WRITE ACTIVATION NO. (TP/PP)
PROCEED WITH INSTALLATION
NO
EXIT
YES
INSTALL

FIG. 5

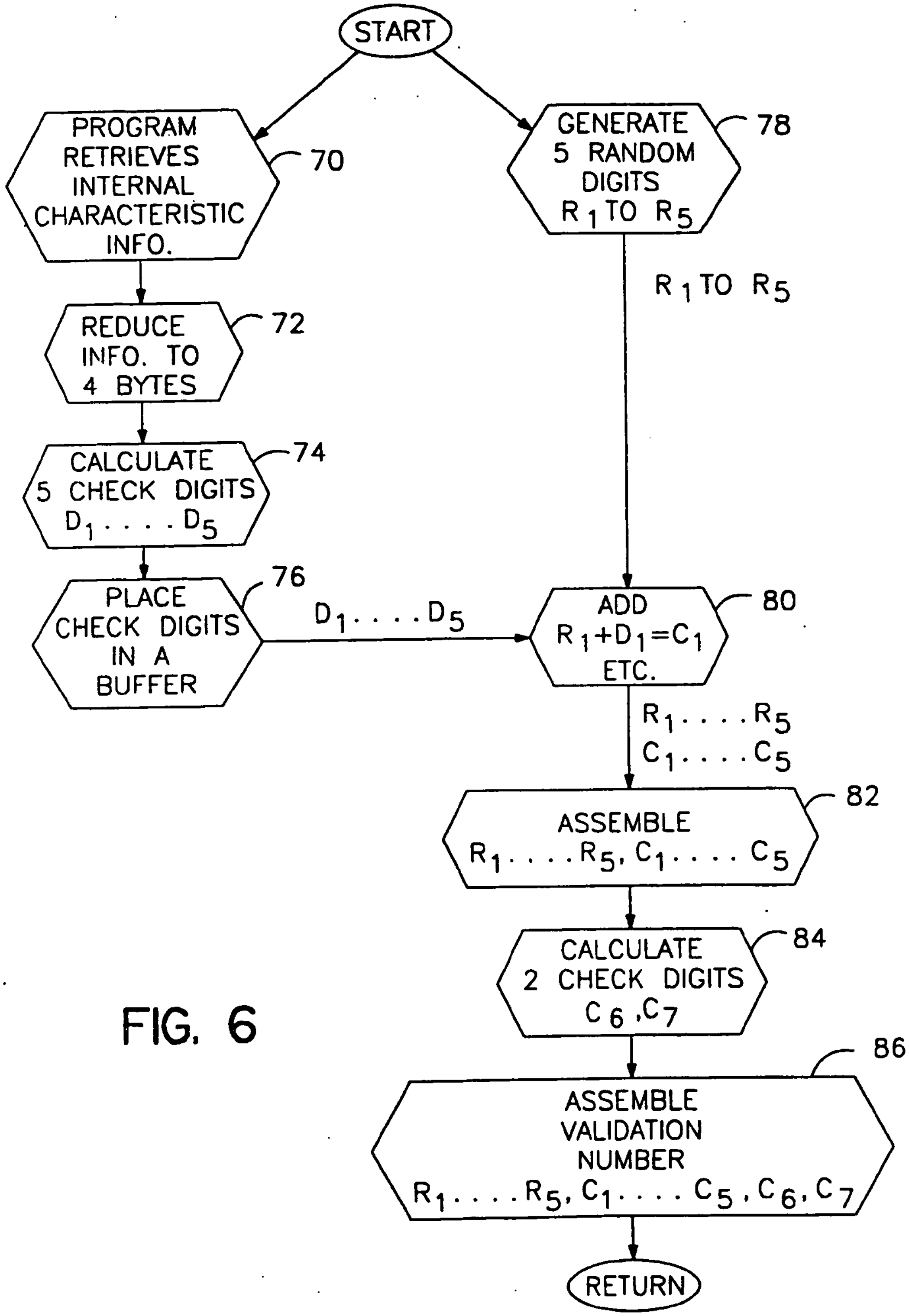

START
PROGRAM RETRIEVES INTERNAL CHARACTERISTIC INFO.
70
REDUCE INFO. TO 4 BYTES
72
CALCULATE 5 CHECK DIGITS $D_1 \ldots D_5$
74
PLACE CHECK DIGITS IN A BUFFER
76
$D_1 \ldots D_5$
GENERATE 5 RANDOM DIGITS R_1 TO R_5
78
R_1 TO R_5
ADD $R_1+D_1=C_1$ ETC.
80
$R_1 \ldots R_5$
$C_1 \ldots C_5$
ASSEMBLE $R_1 \ldots R_5, C_1 \ldots C_5$
82
CALCULATE 2 CHECK DIGITS C_6, C_7
84
ASSEMBLE VALIDATION NUMBER $R_1 \ldots R_5, C_1 \ldots C_5, C_6, C_7$
86
RETURN

FIG. 6

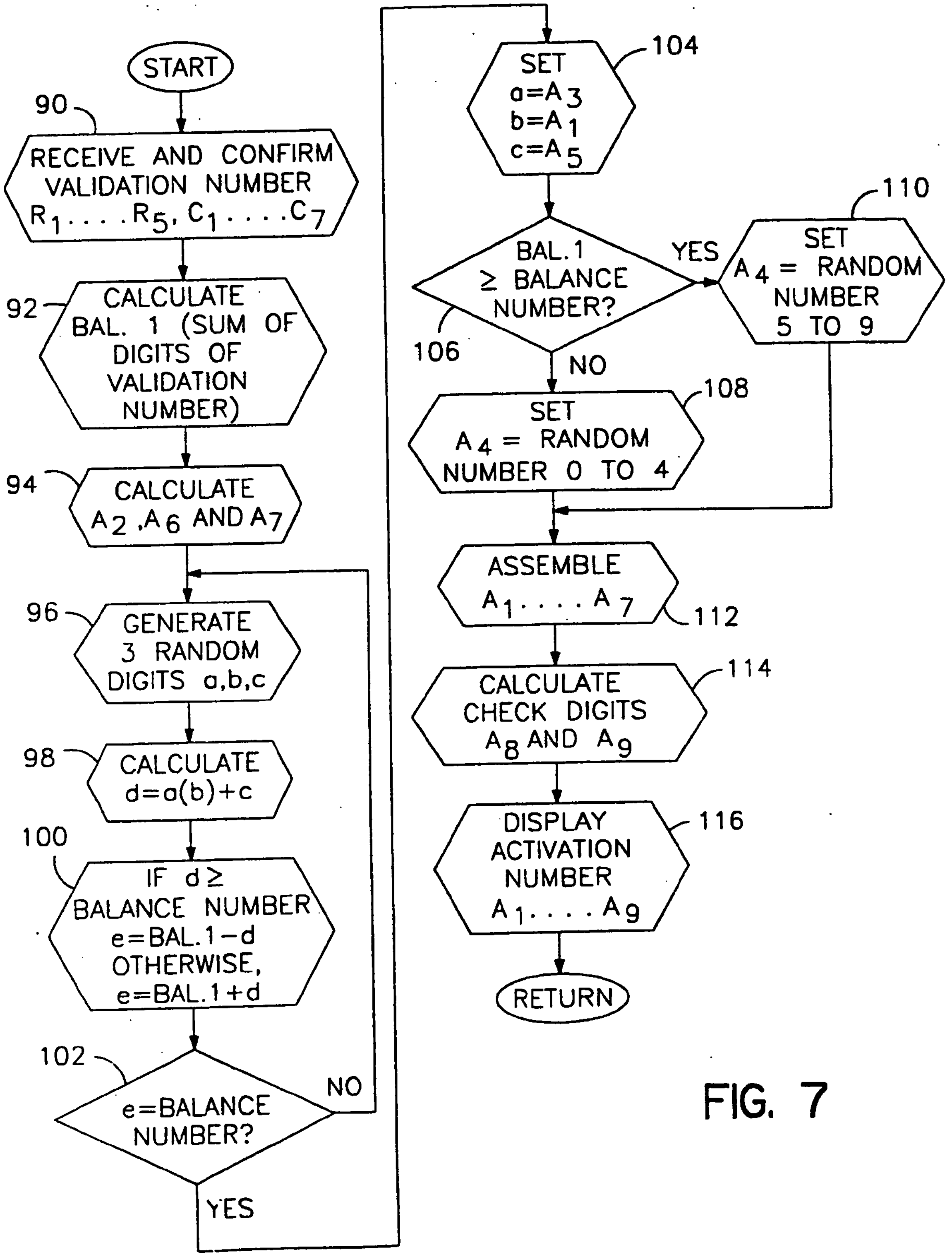

START
90
RECEIVE AND CONFIRM VALIDATION NUMBER $R_1 \ldots R_5, C_1 \ldots C_7$
92
CALCULATE BAL. 1 (SUM OF DIGITS OF VALIDATION NUMBER)
94
CALCULATE A_2, A_6 AND A_7
96
GENERATE 3 RANDOM DIGITS a,b,c
98
CALCULATE $d=a(b)+c$
100
IF $d \geq$ BALANCE NUMBER $e=BAL.1-d$ OTHERWISE, $e=BAL.1+d$
102
e=BALANCE NUMBER?
NO
YES
104
SET $a=A_3$ $b=A_1$ $c=A_5$
106
BAL.1 $\geq$ BALANCE NUMBER?
YES
NO
110
SET $A_4 =$ RANDOM NUMBER 5 TO 9
108
SET $A_4 =$ RANDOM NUMBER 0 TO 4
112
ASSEMBLE $A_1 \ldots A_7$
114
CALCULATE CHECK DIGITS A_8 AND A_9
116
DISPLAY ACTIVATION NUMBER $A_1 \ldots A_9$
RETURN

FIG. 7

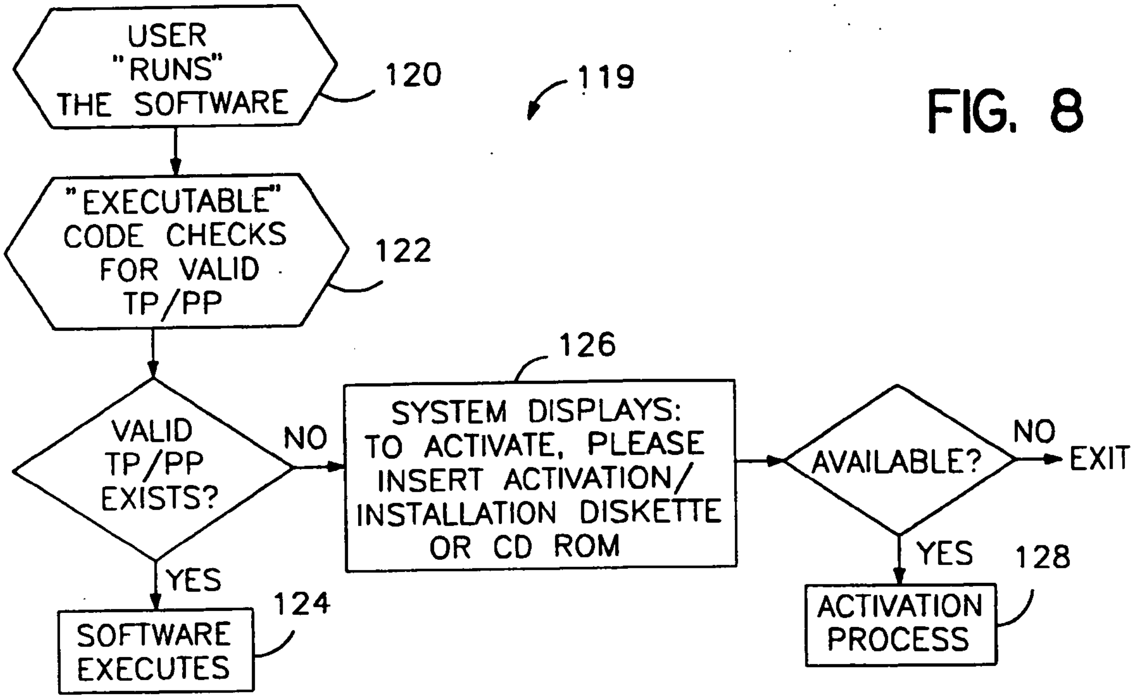

USER "RUNS" THE SOFTWARE
120
119
"EXECUTABLE" CODE CHECKS FOR VALID TP/PP
122
VALID TP/PP EXISTS?
NO
YES
SOFTWARE EXECUTES
124
126
SYSTEM DISPLAYS: TO ACTIVATE, PLEASE INSERT ACTIVATION/ INSTALLATION DISKETTE OR CD ROM
AVAILABLE?
NO
EXIT
YES
128
ACTIVATION PROCESS

FIG. 8

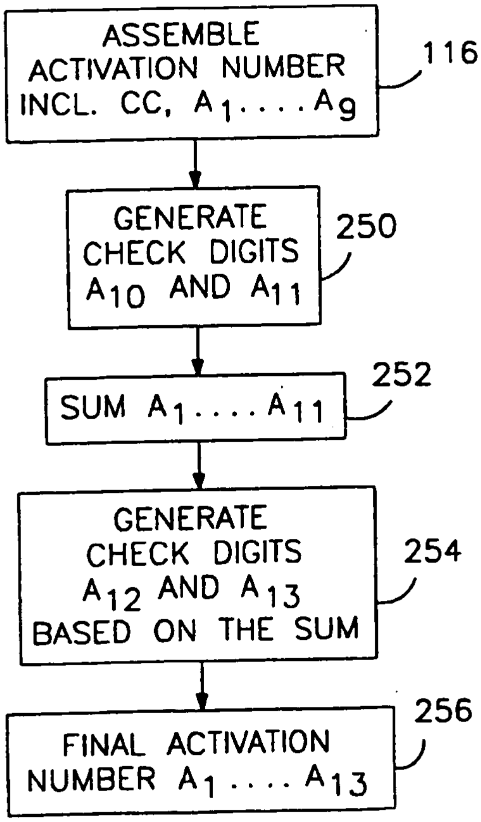

ASSEMBLE ACTIVATION NUMBER INCL. CC, A1....A9
116
GENERATE CHECK DIGITS A10 AND A11
250
SUM A1....A11
252
GENERATE CHECK DIGITS A12 AND A13 BASED ON THE SUM
254
FINAL ACTIVATION NUMBER A1....A13
256

FIG. 17

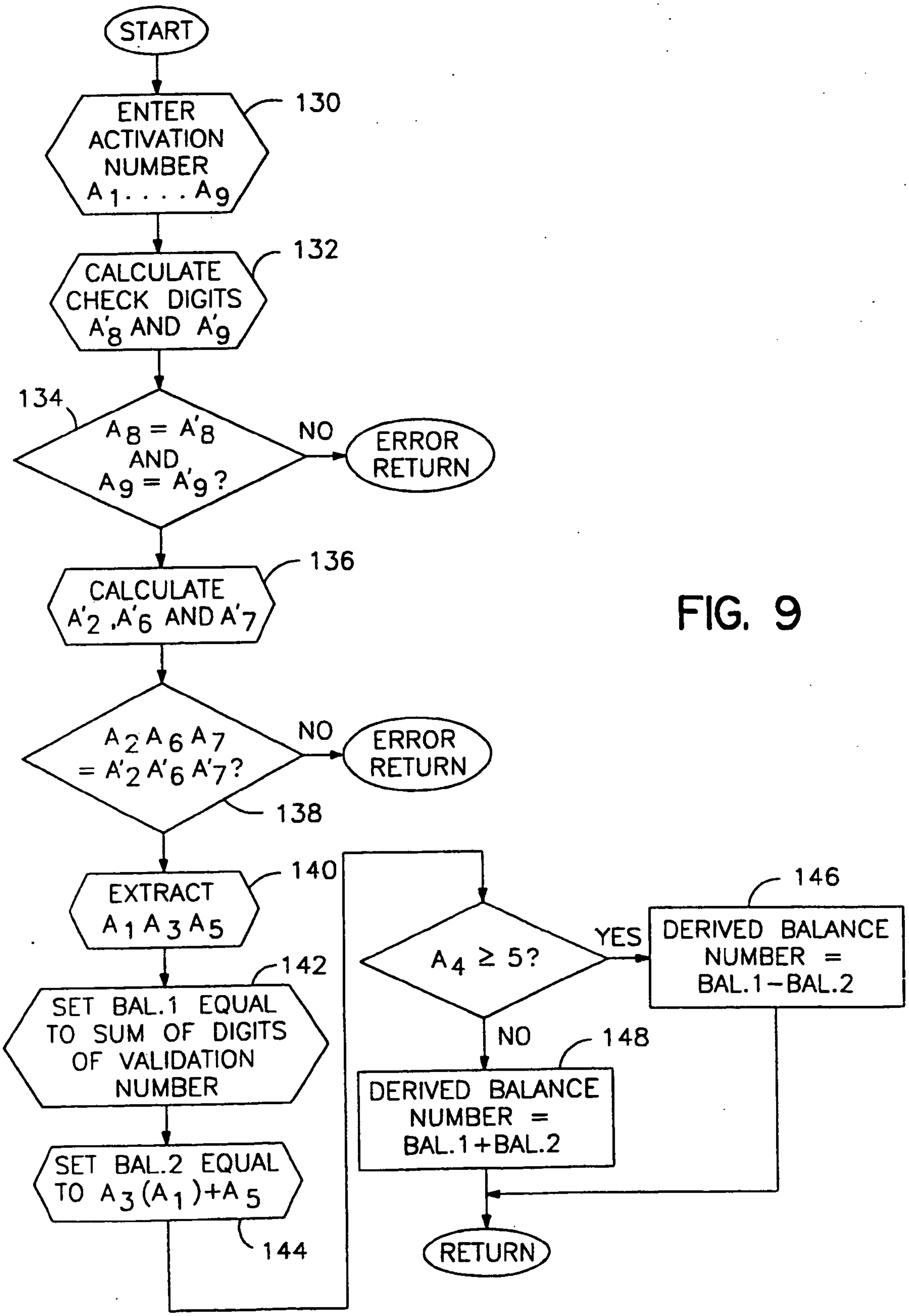

START
ENTER ACTIVATION NUMBER $A_1 \ldots A_9$
130
CALCULATE CHECK DIGITS A'_8 AND A'_9
132
134
$A_8 = A'_8$ AND $A_9 = A'_9$?
NO
ERROR RETURN
CALCULATE A'_2, A'_6 AND A'_7
136
$A_2 A_6 A_7 = A'_2 A'_6 A'_7$?
NO
ERROR RETURN
138
EXTRACT $A_1 A_3 A_5$
140
SET BAL.1 EQUAL TO SUM OF DIGITS OF VALIDATION NUMBER
142
SET BAL.2 EQUAL TO $A_3 (A_1) + A_5$
144
$A_4 \geq 5$?
YES
DERIVED BALANCE NUMBER = BAL.1 − BAL.2
146
NO
DERIVED BALANCE NUMBER = BAL.1 + BAL.2
148
RETURN

FIG. 9

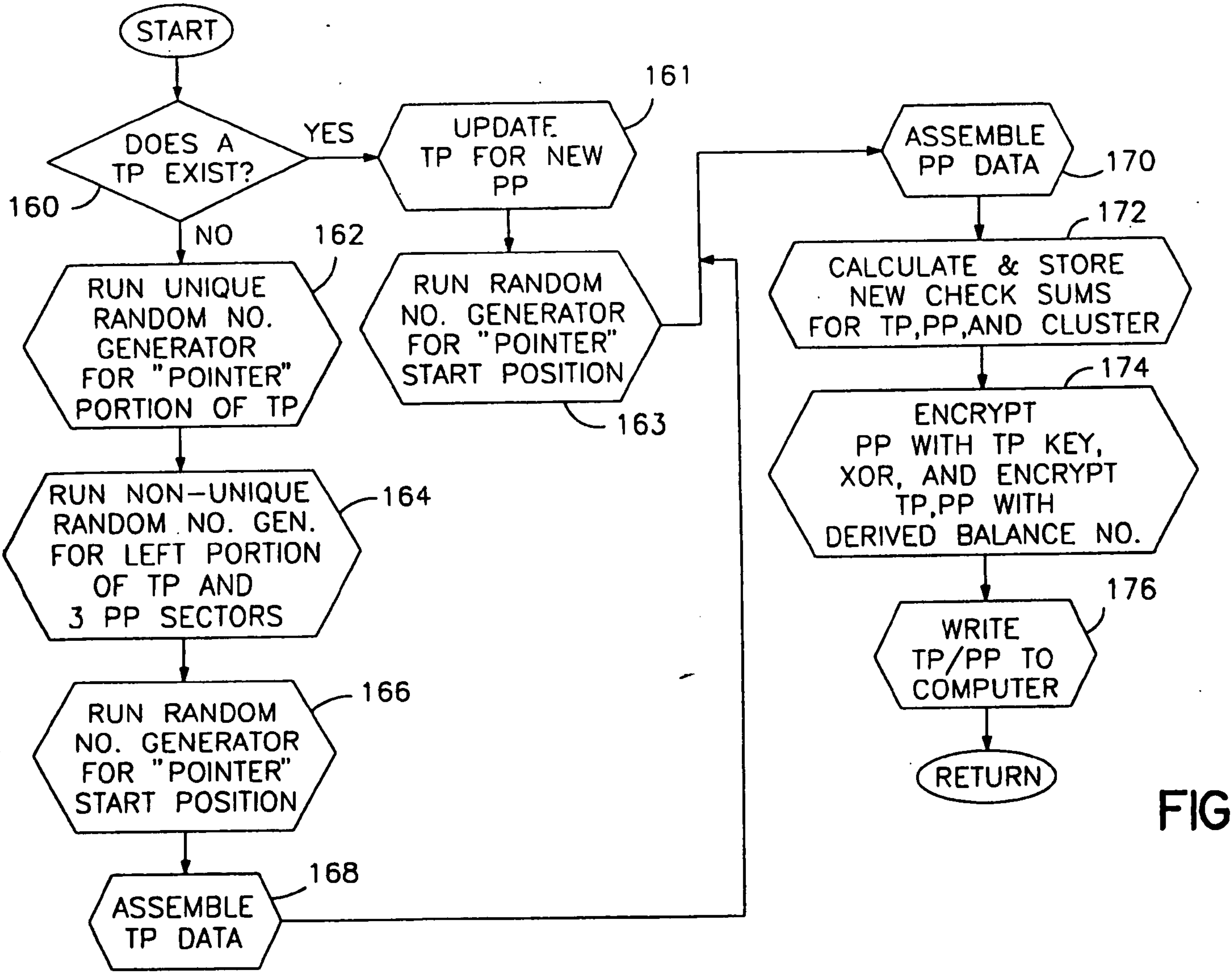

START
DOES A TP EXIST?
160
YES
NO
UPDATE TP FOR NEW PP
161
RUN RANDOM NO. GENERATOR FOR "POINTER" START POSITION
163
RUN UNIQUE RANDOM NO. GENERATOR FOR "POINTER" PORTION OF TP
162
RUN NON-UNIQUE RANDOM NO. GEN. FOR LEFT PORTION OF TP AND 3 PP SECTORS
164
RUN RANDOM NO. GENERATOR FOR "POINTER" START POSITION
166
ASSEMBLE TP DATA
168
ASSEMBLE PP DATA
170
CALCULATE & STORE NEW CHECK SUMS FOR TP,PP,AND CLUSTER
172
ENCRYPT PP WITH TP KEY, XOR, AND ENCRYPT TP,PP WITH DERIVED BALANCE NO.
174
WRITE TP/PP TO COMPUTER
176
RETURN

FIG. 10

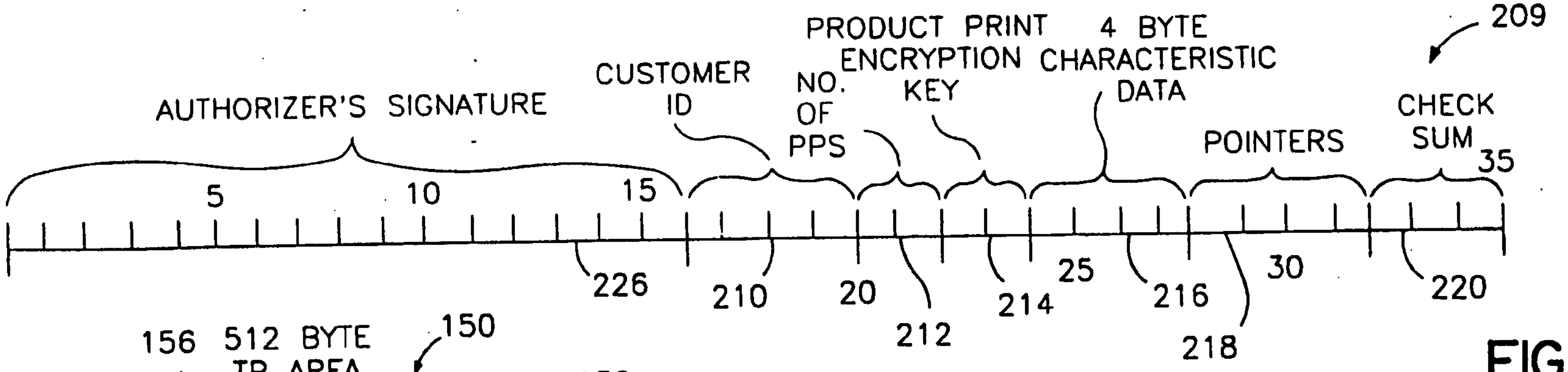


FIG. 11

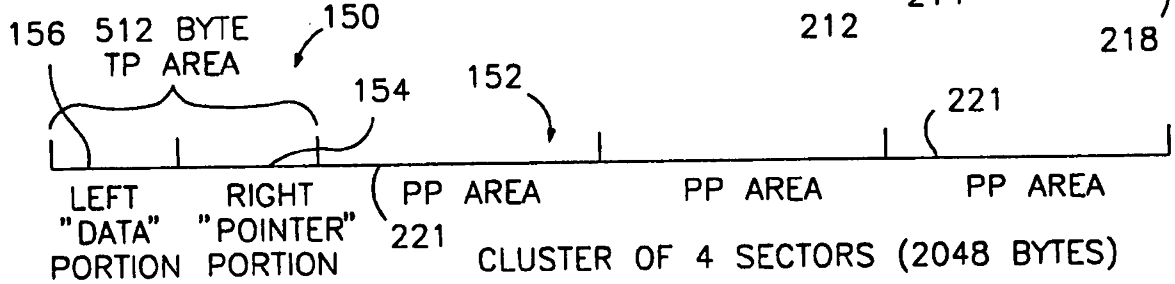


FIG. 13

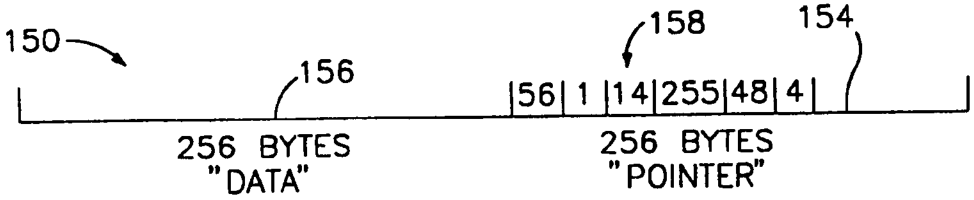


FIG. 14

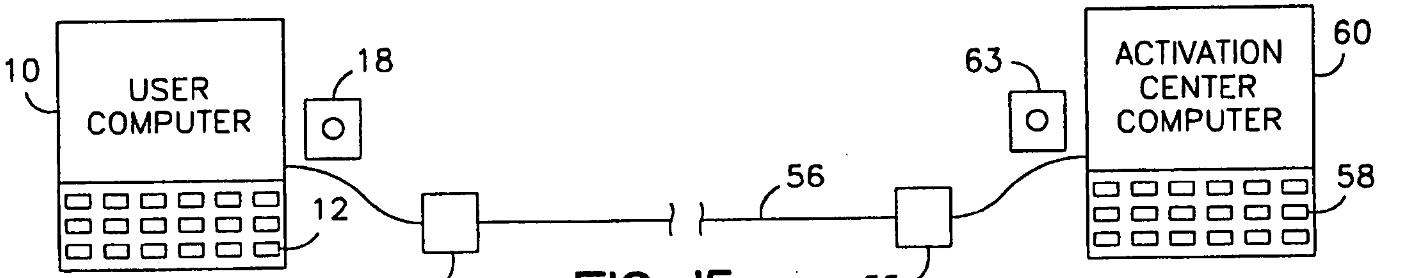


FIG. 15

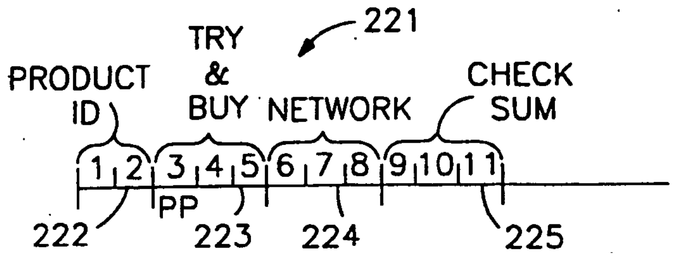

221
PRODUCT ID
TRY & BUY
NETWORK
CHECK SUM
1 2 3 4 5 6 7 8 9 10 11
PP
222
223
224
225

FIG. 12

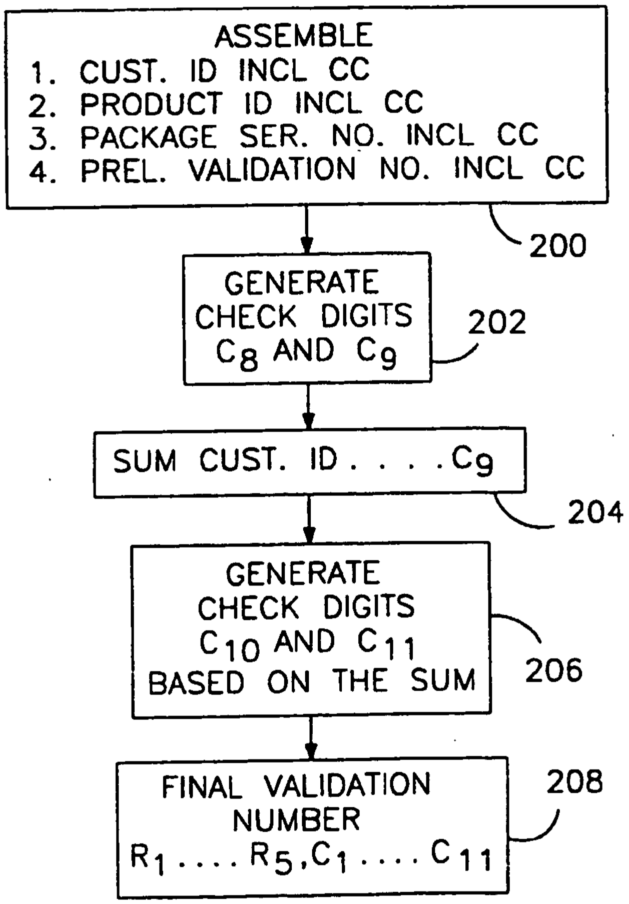

ASSEMBLE
1. CUST. ID INCL CC
2. PRODUCT ID INCL CC
3. PACKAGE SER. NO. INCL CC
4. PREL. VALIDATION NO. INCL CC
200
GENERATE CHECK DIGITS C_8 AND C_9
202
SUM CUST. ID . . . . C_9
204
GENERATE CHECK DIGITS C_{10} AND C_{11} BASED ON THE SUM
206
FINAL VALIDATION NUMBER $R_1 \ldots R_5, C_1 \ldots C_{11}$
208

FIG. 16

INTERNATIONAL SEARCH REPORT

International Application No
PCT/CA 95/00354

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,X	US,A,5 337 357 (CHOU ET AL) 9 August 1994 see abstract; figure 1 see column 2, line 57 - column 3, line 17	1,5,10, 12,14,20
P,Y		6,8,9, 13,21

X	US,A,4 796 220 (WOLFE) 3 January 1989 cited in the application see abstract; figures 1-3 see column 3, line 41 - column 4, line 48 see column 6, line 4 - column 8, line 68	1,2,5
A		8,13,15, 17,19, 21,24,26

	-/--	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

1

Date of the actual completion of the international search 11 October 1995	Date of mailing of the international search report 18. 10. 95
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+ 31-70) 340-3016	Authorized officer Powell, D

INTERNATIONAL SEARCH REPORT

International Application No
PCT/[illegible] 95/00354

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO,A,94 07204 (UNILOC) 31 March 1994 see abstract; figure 8 see page 7, line 4 - page 8, line 28 see page 22, line 14 - page 23, line 9	6,8,13, 21
A		1,2

Y	US,A,5 023 907 (JOHNSON ET AL) 11 June 1991 see abstract; figure 2 see column 3, line 66 - column 5, line 41	9
A		8,21

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/CA 95/00354

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
US-A-5337357	09-08-94	CA-A-	2120816	18-12-94
		EP-A-	0636962	01-02-95
US-A-4796220	03-01-89	NONE		
WO-A-9407204	31-03-94	AU-B-	4811393	12-04-94
		CA-A-	2145068	31-03-94
		CN-A-	1103186	31-05-95
US-A-5023907	11-06-91	NONE		